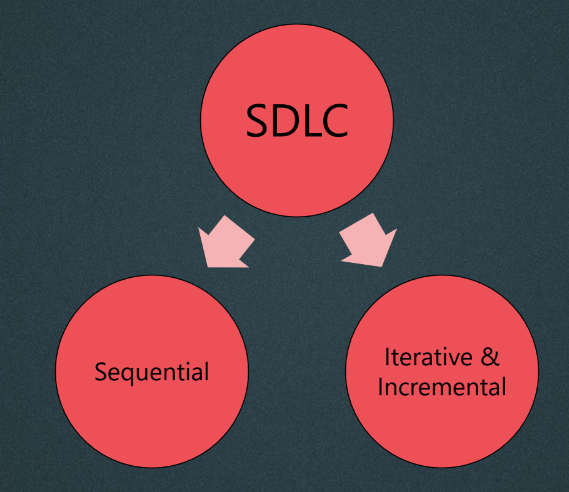
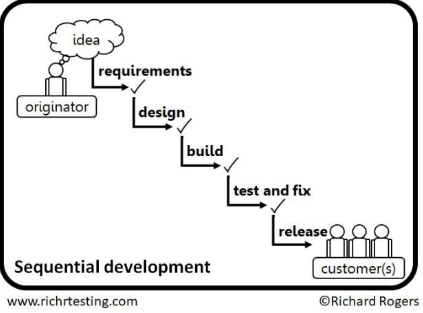
Kurs Linki: [The Complete 2023 Software Testing Bootcamp | Udemy](https://www.udemy.com/course/testerbootcamp/)

**SDLC (System Development Life Cycle)**

🡺 SDLS iki kısma ayrılır. Birincisi sıralı (**Sequential**) geliştirme, ikincisi yinelemeli (**Iterative**) ve artımlı (**Incremental**) geliştirmedir.



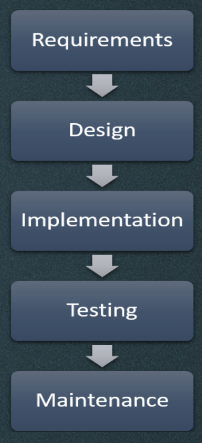
**1)** Sıralı geliştirme:



🡺 Sıralı geliştirmede adımlar birbirini takip eder. Teorik olarak bir adımın başlaması için önceki adımın bitmesi gerekir. Bu duruma göre adımlar arasında çakışma olmaz. Uygulama kısmında ise projenin bir kısmı tasarlarken, önceden tasarlanan kısmın testi yapılabilir. Aynı şekilde projeyi geliştirme devam ederken müşteri farklı bir özellik istediğinde önceki adımlara geri dönülebilir. Bu şekilde istisnalar bulunabilir.

🡺 Sıralı geliştirme iki kısma ayrılır. Şelale (**Waterfall**) ve **V-Model**.

**1.1)** Şelale (Waterfall) geliştirme:



🡺 Bu modelde test işlemleri, bütün işlemler bittikten sonra başlar.

🡺 Şelale modeli denmesinin sebebi, şelalelerde akan suyun bir daha geri dönmemesinden gelmektedir. Uygulama gereksinimleri toplanır, tasarlanır, uygulanır, test edilir ve yayınlanır. Eğer müşteri farklı bir özellik daha eklemek isterse, o zaman projeye bakım (**Maintenance**) yapılır.

**1.2)** V-Model geliştirme:

metin içeren bir resim

Açıklama otomatik olarak oluşturuldu

🡺 V-Model geliştirmede gereksinimler Kullanıcı gereksinimleri ve yazılım gereksinimleri (veya teknik gereksinimler) olarak ikiye ayrılır. Tasarım kısmı, yüksek seviye tasarım ve düşük seviye tasarım olarak iki bölüme ayrılır. Projeyi geliştirmeye sol taraftan başlanır. V-Model geliştirmede ilk 5 adım şelale geliştirmenin ilk 3 adımı ile aynıdır.

🡺 V- Model geliştirmede test kısmı 4 bölüme ayrılır. Bunlar birim testi, entegrasyon testi, sistem testi ve kabul testidir.

**2)** Yinelemeli (Iterative) ve artımlı (Incremental) geliştirme:

metin, farklı içeren bir resim

Açıklama otomatik olarak oluşturuldu

🡺 Yinelemeli (**Iterative**) geliştirmede proje iterasyonlara (Birbirini takip eden işlemler) bölünür. Mesela yukarıdaki örnekte, ressam ilk olarak resmi sadece kara kalemle çizmiş daha sonra adımlar ilerleyince renklendirmiş.

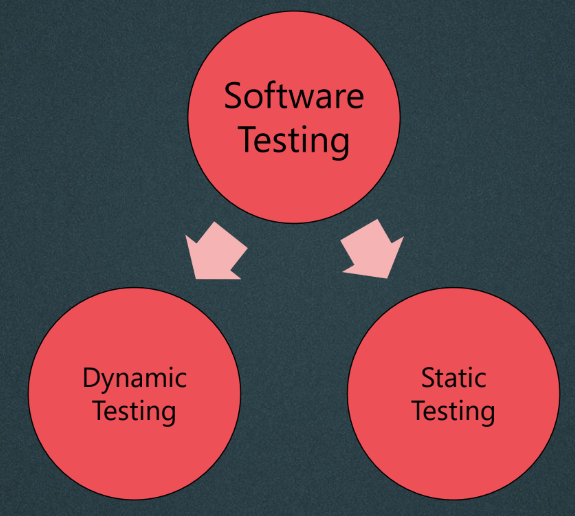
🡺 Artımlı (**Incremental**) geliştirmede proje bölümlere ayrılır. Bir bölümün gereksinimleri toplanır, tasarlanır, uygulanır ve test edilir. Daha sonra diğer bölüme geçilir.

🡺 Projemizde aynı anda hem artımlı geliştirmeyi hem de yinelemeli geliştirmeyi kullanabiliriz. Yani artımlı geliştirmede ayırdığımız bölümleri iterasyonlara ayırıp geliştirmemizi yapabiliriz. Bu duruma en iyi örnek **Scrum’lardır**.

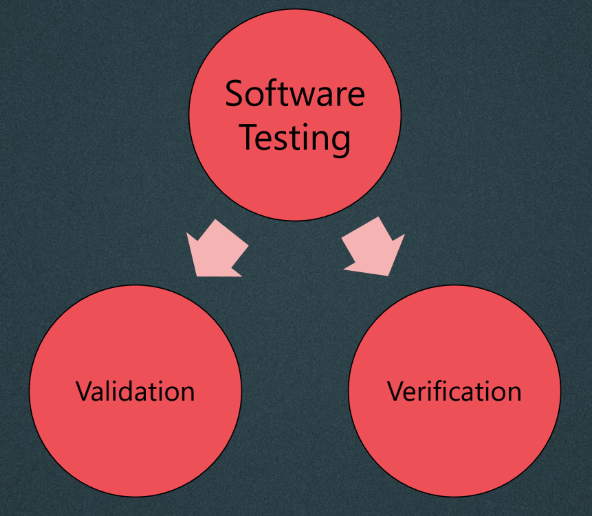
***NOT:*** Scrum nedir 🡪 [SCRUM NEDİR?. Zaman içerisinde projelerin daha büyük ... - Medium](https://medium.com/@secilcor/scrum-nedi%CC%87r-6a4326951dd8)  
***NOT:*** Scrum nedir video anlatım 🡪 [Introduction to Scrum - 7 Minutes - YouTube](https://www.youtube.com/watch?v=9TycLR0TqFA)

**Software Testing**

🡺 Yazılım testi, Dinamik (**Dynamic**) test ve Statik (**Static**) test olarak iki kısma ayrılabilir. Dinamik testte yazılımı çalıştırıp test ederiz. Statik testlerde yazılımı çalıştırmayız. Gereksinimleri gözden geçirebiliriz, tasarımı gözden geçirebiliriz, kodumuzu gözden geçirebiliriz kısacası her türlü gözden geçirmeyi yapabiliriz.



🡺 Ayrıca yazılım testi, Doğrulama (**Validation**) ve Onaylama (**Verification**) olarak 2 kısma ayrılabilir. Doğrulama (**Validation**) doğru ürünü oluşturduğumuz anlamına gelir. Kullanıcı bu ürünü kullandığında memnun olacak. Onaylama (**Verification**) ürünü doğru bir şekilde oluşturduğumuz anlamına gelir.



🡪 Test süreci, yazılımı test etmek için uyguladığımız adımlardır. Test süreci 3 adıma ayrılır. Plan(Planlama), Design(Tasarım) ve Execution(Yürütme). Yazılım testi için tek bir evrensel test süreci yoktur.

diyagram içeren bir resim

Açıklama otomatik olarak oluşturuldu

🡪 2018 ISTQB müfredatına göre test süreci 7 faaliyetten oluşur. İlk olarak test planımızı yazarız. Daha sonra projede ilerlerken, ilerlememizi izlemeye başlarız. Gerçek ilerlememizi, planlama aşamasında yazdığımız beklenen ilerleme ile karşılaştırırız. Eğer bir fark var ise Control(Kontrol) faaliyetlerini gerçekleştiririz. Daha sonra gereksinimlerimizi okuyarak, analiz ederiz. Bu aşamadan sonra test senaryoları yazmaya başlarız. Bir sonraki adımda ise testlerimizi uygulamaya başlarız. Daha sonra farklı platformlarda testleri yürütürüz, hataları buluruz, sonuçları raporlarız, yeniden test ederiz ve regresyon testleri gerçekleştiririz. En son adımda ise test işlemini tamamlarız, test ilerleme raporları ve test özet raporları yazarız.

dikdörtgen içeren bir resim

Açıklama otomatik olarak oluşturuldu

**Test Levels**

🡪 4 Ana test seviyesi vardır.

**1) Unit Test – Component Test (Birim testi – Modül (Bileşen) testi):** Ayrı olarak test edilenleri test eder. Kod modülleri, Kod sınıfları vb. Çoğu zaman bu test türünden sorumlu olan kişi geliştiricidir.

**2) Integration Test (Entegrasyon testi):** Entegrasyon testi, bileşenler veya sistemler arasındaki etkileşimlere odaklanır. İki tür entegrasyon testi vardır. Bunlar Component Integration (Bileşen entegrasyon) ve System Integration (Sistem entegrasyon) testleridir. Bileşen entegrasyon testi, bileşenler arasındaki entegrasyona odaklanır. Bu test türü de geliştirici tarafından gerçekleştirilir. Sistem entegrasyon testi ise backend, database frontend vb sistemlerin birbirine entegre edilmesine denir. Sistem entegrasyon testini, test uzmanı gerçekleştirir.

**3) System Test (Sistem testi):** Sistem testi, sistemi bir bütün olarak test etmeye denir. Bu test uzmanları için en önemli test seviyesidir. Bu seviyede uygulamamızı gerçek ortama benzer bir ortamda test etmeye çalışırız. Sistem testi sırasında mümkün oluğunca çok hata bulmaya çalışırız. Bu test seviyesi, test uzmanının sorumluluğundadır.

**4) Acceptance Test (Kabul testi):** Sistem testini gerçekleştirip hataları bulduktan sonra, kabul testini uyguluyoruz. Bu testin amacı, uygulamamızın doğru performans gösterip göstermediğinden emin olmaktır. Hata bulmak bu test seviyesinin ana amacı değildir. Bu test seviyesi, çoğu zaman kullanıcılar ve paydaşlar tarafından yapılır. Alpha (Alfa) testi ve Beta testi olarak adlandırılan bir kabul testimiz vardır. Alfa testi, müşterileri şirkete getirdiğimiz ve tesislerimizde test etmelerini sağladığımız anlamına gelir. Beta testi ise müşterilerimizin uygulamamızı kendi evlerinde test etmelerini sağladığımız anlamına gelir.

Testing Types